BAB II
LANDASAN TEORI

Pada bagian ini akan dijelaskan mengenai beberapa teori penunjang yang berhubungan dengan pokok bahasan dalam tugas akhir ini. Seperti konsep dasar *High Availability*, *failover,* replikasi, *cloud computing*.

2.1 Fault, Failures dan Outages

Secara konsep, fault bisa di artikan sebagai sebuah tingkah laku sistem yang tidak biasa. Sedangkan *failures* adalah sebuah fault yang terlihat oleh *end user*. Sistem yang menerapkan konsep *high availability* akan langsung menangani apabila *fault* mampu *failures* (Engelmann, 2006).

Kegagalan atau *fault* secara tipikal di katagorikan sebagai berikut (Gray, 1991):

1) Kegagalan Hardwere : kegagalan fungsionalitas *device* pada sistem
2) Kegagalan Desain : kegagalan pada *softwere* dan desain *hardwere*
3) Kegagalan Operasional: kegagalan yang terjadi akibat kesalahan operasi dan *maintenace*.
4) Kegagalan Lingkungan: kegagalan akibat kebakaran, banjir, gempa bumi, listrik padam, dan sabotase.

Outages, adalah istilah yang di pakai untuk mendeskripsikan segala jenis kalainan yang terjadi pada sistem, baik yang di harapkan maupun tidak diharapakan. Segala jenis *fault* dan *failures* bisa di katagorikan sebagai unplanned *outages* atau kelainan yng tidak di harapkan (Englemann, 2006) [7].

2.2 Cloud Computing

Cloud computing atau komputasi awan adalah sebuah teknologi yang memanfaatkan layanan internet atau intranet (Local) sebagai pusat *server* yang bersifat virtual dengan tujuan mempermudah pemeliharaan data dan aplikasi. Keberadaan *cloud computing* akan sangat memberi perubahan dalam cara kerja

suatu sistem teknologi informasi dalam suatu organisasi. Karena *cloud computing* melalui sebuah konsep virtualisasi, standarisasi dan fitur mendasar yang dapat mengurangi biaya Teknologi Informasi (TI) [8]. Secara umum arsitektur komputasi awan terdiri dari *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) dan *Software as a Service* (SaaS) [9].

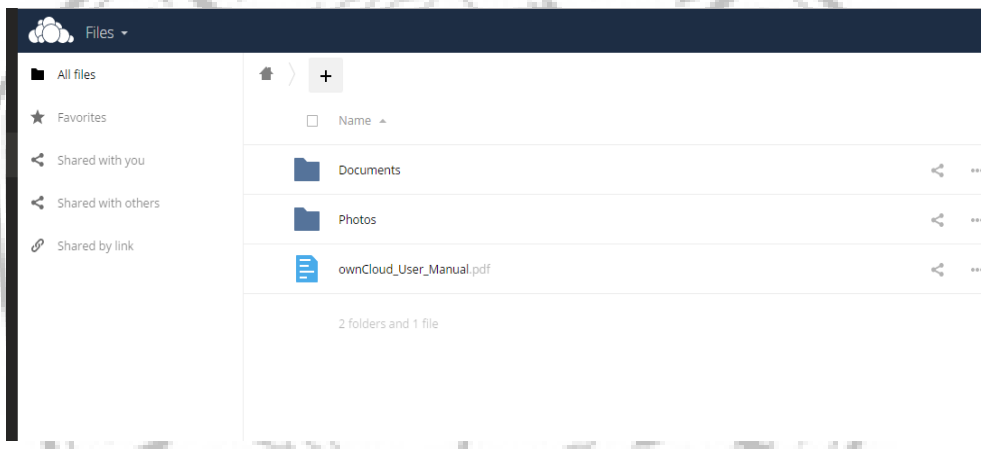
1. *Cloud Software as a Service* (SaaS). Adalah hak yang diberikan kepada pengguna untuk menggunakan aplikasi penyedia yang beroperasi pada infrastruktur *cloud*. Aplikasi yang di sediakan dapat diakses melalui perangkat klien dengan antarmuka seperti *web browser*. Pengguna tidak dapat mengelola atau mengendalikan infrastruktur *cloud* yang mendasar termasuk jaringan, *server*, sistem operasi yang di gunakan, besar penyimpanan, dengan kemungkinan pengecualian hanya pada pengaturan dari pengolahan data pengguna.
2. *Cloud Platform as a Service* (PaaS). Adalah hak yang diberikan kepada pengguna untuk menyebarkan sebuah aplikasi yang dibuat pengguna untuk infrastruktur *cloud computing* menggunakan bahasa pemrograman dan peralatan yang didukung oleh penyedia layanan atau provider. Pengguna tidak mengelola atau mengendalikan infrastruktur *cloud* yang mendasar termasuk jaringan, *server*, sistem operasi yang di gunakan, atau besar penyimpanan, akan tetapi memiliki control penuh atas aplikasi yang disebarkan dan memungkinkan aplikasi dilakukan hosting konfigurasi.
3. *Cloud Infrastructure as a Service* (IaaS). Adalah hak yang diberikan kepada pengguna untuk dapat memproses, menyimpan konfigurasi, menata jaringan, dan melakukan perubahan sumberdaya lain yang di butuhkan, dimana pengguna dapat menyebarkan dan menjalankan suatu perangkat lunak secara bebas, dapat meliputi sistem operasi dan aplikasi. Pengguna tidak mengelola atau mengendalikan infrastruktur *cloud* yang mendasar akan tetapi memiliki kontrol atas suatu sistem operasi, penyimpanan, aplikasi yang akan disebarkan.

Cloud Storage merupakan salah satu bentuk dari *cloud computing* namun terpusat pada media penyimpanan yang dalam pengaksesannya memerlukan

jaringan internet maupun intranet. Di dalam *cloud storage* ini seseorang di beri hak untuk melakukan akses ke *resource* yang di miliki oleh penyedia layanan *cloud storage*. Sehingga pengguna tidak harus membawa *storage* dengan fisik besar seperti *hardisk* dan alat penyimpanan lain. Karena *cloud storage* sudah menyediakan layan penyimpanan dan lebih efisien waktu dan tempat di bandingkan penyimpanan fisik.

2.3 Owncloud

Owncloud merupakan aplikasi *open source* yang juga mendukung teknologi *cloud computing*. Owncloud menyajikan layanan cloud storage seperti *dropbox* dan *google drive*, yaitu layanan penyimpanan dan pengambilan file secara *online* pada jaringan *cloud computing*.



Gambar 1.1 Interface Owncloud

2.4 High Availability

High Availability adalah kemampuan sebuah sistem untuk melanjutkan fungsinya tanpa intrupsi dalam kurun waktu yang cukup lama dari kemampuan komponen yang seharusnya. Salah satu tujuan paling mendasar dari *high availability* adalah menghindari kegagalan sebuah *single point* untuk mencapai kelanjutan operasional, redudancy dan kemampuan *failover*.

2.5 Failover

Failover adalah peralihan ke sebuah perangkat komponen atau *server* cadangan saat *server* utama mengalami gangguan. Dengan menggunakan *failover*

kita dapat mengakses data yang di simpan pada saat *server* utama mengalami gangguan.

Failover clustering merupakan suatu bentuk *failover* yang menyediakan *high availability server* ketika jika terjadi kegagalan pada sebuah sistem *hardware* seperti *power supply* mati sehingga *server* mati total, maka *server* yang menjadi anggota *cluster* lain yang akan mengambil alih semua fungsi dari *server* yang mengalami kegagalan, sehingga pengguna tidak mengetahui jika terjadi kegagalan pada *server,* karena proses yang dilakukan pada server yang mengalami kegagalan akan dilanjutkan oleh *server backup. Failover* memiliki beberapa metode dalam menangani kegagalan sistem atau komponen.

1. *Manual Masking*: Dibutuhkan campur tangan manusia untuk melakukan aktifitas komponen cadangan untuk menggantikan komponen utama.
2. *Cold Stanby*: Dibutuhkan otomoatis yang menggantikan komponen utama dengan komponen cadangan. Ketika terjadi interupsi pada *service* dan komponen utama tidak bias berfungsi. Solusi ini biasanya di pakai untuk redudansi *hardwere*, bukan *softwere*.
3. *Warm Stanby* merupakan salah satu metode *failover* dimana metode Ini membutuhkan replikasi dan *failover* otomatis untuk meggantikan komponen atau *service* utama yang gagal. Solusi ini di pakai untuk melalukan redudansi *hardwere* dan *softwere*. Di dalam metode ini *server* utama secara teratur akan melakukan replikasi ke *server* cadangan. Ketika *server* utama mengalami kegagalan *server* cadangan akan menggantikan peran *server* utama berdasarkan replikasi terakhir yang tercatat dari *server* utama sebelum *server* utama mengalami kegagalan.
4. *Hot Standby* merupakan salah satu mode *failover* di mana replikasi penuh terhadap semua komponen utama dalam komponen cadangan, serta prosedur *failover* otomatis dari komponen utama kedalam komponen cadangan. Dalam hal ini, komponen utama tidak sepenuhnya gagal, namun hanya beberapa *service* saja yng mengalami interupsi. Komponen cadangan terus menerus menerima update dan replikasi dari komponen utama. Ketika

terjadi interupsi *service* dalam komponen utama, komponen cadangan akan menangani *service* yang mengalami interuspsi berdasarkan data yang di dapat dari komponen utama yang sedang berjalan.

2.6 Replikasi

Replikasi adalah sebuah teknik proses untuk *copy* dan mengirim data atau beberapa *objek-objek* dari satu *server database* ke *server database* lain dan melakukan proses sinkronisasi antara *server database* sehingga konsistensi data yang di simpan dapat terjamin. Replikasi bisa di artikan juga sebagai teknik penggandaan *database* dan pengelolaan suatu *objek-objek Database* dalam suatu jaringan komputer yang dapat membentuk suatu sistem *database* terdistribusi untuk menjaga setiap konsistensi data secara otomatis. (Purwanto, Eddy S.Kom. 2012)

Replikasi memiliki beberapa macam metode pada saat sinkronisasi data:

1. Metode Single Master Replicated

   Dengan metode ini, satu komputer berfungsi sebagai server master dan yang lainnya di fungsikan sebagai *slave*. Saat prosesnya, *server* akan dapat melakukan *read* dan *write* ke dalam *database*. Sedangkan *slave*, hanya akan melakukan *read* saja kedalam *database* tersebut. Apabila ada perubahan data pada *master*, maka secara otomatis datayang berada pada *slave* akan berubah sesuai dengan data yang ada pada di *master*. Tetapi sebalikanya jika kita melakukan perubahan data pada *slave*, data yang terdapat pada master tidak akan berubah.

2. Metode Multiple Master Replicated

   Dengan metode ini, satu komputer berfungsi sebagai server master dan yang lainnya di fungsikan sebagai *slave*. Saat prosesnya, kedua *server* tersebut melakukan tugas yang sama di mana ketika *server* utama mengalami perubahan *database* makan akan terjadi pula di *server slave*. Dan sebalikanya juga *server slave* mengalami perubahan makan *server* utama mengalami perubahan juga.